

ARMY RESEARCH LABORATORY



Truncated Newton-Raphson Methods for Quasicontinuum Simulations

by Yu Liang, Ramdev Kanapady, and Peter W. Chung

ARL-TR-3791

May 2006

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory

Aberdeen Proving Ground, MD 21005-5067

ARL-TR-3791

May 2006

Truncated Newton-Raphson Methods for Quasicontinuum Simulations

Yu Liang and Ramdev Kanapady
University of Minnesota

Peter W. Chung
Computational and Information Sciences Directorate, ARL

REPORT DOCUMENTATION PAGE				<i>Form Approved OMB No. 0704-0188</i>
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.				
1. REPORT DATE (DD-MM-YYYY) May 2006	2. REPORT TYPE Final			3. DATES COVERED (From - To) 2003–2005
4. TITLE AND SUBTITLE Truncated Newton-Raphson Methods for Quasicontinuum Simulations			5a. CONTRACT NUMBER DAAD19-01-2-0014	
			5b. GRANT NUMBER	
			5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Yu Liang,* Ramdev Kanapady,* and Peter W. Chung			5d. PROJECT NUMBER 4UH7AC	
			5e. TASK NUMBER	
			5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) U.S. Army Research Laboratory ATTN: AMSRD-ARL-CI-HC Aberdeen Proving Ground, MD 21005-5067			8. PERFORMING ORGANIZATION REPORT NUMBER ARL-TR-3791	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)			10. SPONSOR/MONITOR'S ACRONYM(S)	
			11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.				
13. SUPPLEMENTARY NOTES *Department of Mechanical Engineering, University of Minnesota, Minneapolis, MN 55455				
14. ABSTRACT The quasicontinuum method provides an efficient way to simulate the mechanical response of relatively large crystalline materials at zero temperature by combining continuum and atomistic approaches. Unconstrained optimization constitutes the key computational kernel of this method. The efficiency of the techniques for minimization depends on both the time needed to evaluate the energy expression and the number of iterations needed to converge to the minimum. In this research, we report the effectiveness of the truncated Newton-Raphson method and quasi-Newton method with low-rank Hessian update strategy that are evaluated against the full Newton-Raphson and preconditioned nonlinear conjugate gradient implementation available at qcmethod.com. Results of illustrative examples mainly focus on the number of minimization iterations to converge and CPU time for the two-dimensional nanoindentation and shearing grain boundary problems.				
15. SUBJECT TERMS quasicontinuum, unconstrained optimization, Newton-Raphson method, preconditioned conjugate gradient, nonlinear conjugate gradient, nanoindentation, solver				
16. SECURITY CLASSIFICATION OF: a. REPORT UNCLASSIFIED b. ABSTRACT UNCLASSIFIED c. THIS PAGE UNCLASSIFIED		17. LIMITATION OF ABSTRACT UL	18. NUMBER OF PAGES 38	19a. NAME OF RESPONSIBLE PERSON Peter W. Chung
				19b. TELEPHONE NUMBER (Include area code) 410-278-6027

Contents

List of Figures	v
List of Tables	v
Acknowledgments	vi
1. Introduction	1
2. Quasicontinuum Method	2
2.1 Energy Framework	2
2.2 Atomistic Framework.....	4
2.3 Continuum Framework.....	5
2.4 Coupling Atomistic and Continuum.....	6
2.5 Formulation of Gradient and Hessian	7
2.5.1 Analytic Formulation of Gradient	7
2.5.2 Analytic Formulation of Hessian	8
3. Solvers for Unconstrained Optimization	9
3.1 Nonlinear Conjugate Gradient Method	9
3.1.1 Line Search Algorithm	10
3.1.2 Backtracking.....	10
3.2 NR Methods	11
3.2.1 Numerical Formulation of Hessian	11
3.2.2 Low-Rank Update Method.....	12
3.2.3 Convergence of the NR Method Based on Iterative Solver	12
3.2.4 Implementation of the NR Method	14
4. Numerical Experiments and Results	15
4.1 Shearing Grain Boundary	16
4.2 Simulation of Nanoindentation	18
5. Conclusions	20

6. References	22
Appendix. Supporting Derivations	25
Distribution List	28

List of Figures

Figure 1. (a) Iterative process of quasicontinuum formulation and (b) local and nonlocal framework: black-filled circles are interface atoms shared by local and nonlocal regions; gray-filled circles are atoms selected as elemental nodes in local region; white-filled circles in the local region are formulated using kinematic constraints.....	3
Figure 2. CB rule which transforms the reference lattice into homogeneously deformed (unified deformation gradient tensor F) lattice.....	4
Figure 3. Numerical methods to formulate Hessian matrix.....	12
Figure 4. Sliding of grain boundary: (a) 1st load step, (b) 4th load step, (c) 7th load step, and (d) 11th load step.....	16
Figure 5. Number of minimization iterations for NCG, NR, and TNR for shear grain boundary problem for 10th load step.....	17
Figure 6. CPU time for NCG, NR, and TNR for shear grain boundary problem.....	17
Figure 7. Time cost per minimizing iterations for TNR and NCG and number of iteration of linear conjugate gradient for the TNR for shear grain boundary problem.	17
Figure 8. Nanoindentation problem: (a) 15th load step, (b) 20th load step, (c) 25th load step, and (d) 30th load step.....	18
Figure 9. Performance comparison of the NCG, the NR, and QNR based rank-2 update Hessian for 10th load step of nanoindentation problem.	19
Figure 10. Convergence by nonlinear iteration for nanoindentation: (a) initial load step; (b) stage 1 of 1st load step; (c) 2nd load step; (d) 3rd load step; and (e) 4th load step; NR-PCG = TNR.....	19
Figure 11. Convergence by timing cost (s) for nanoindentation: (1) initial load step; (2) stage 1 of 1st load step; (3) stage 2 of 1st load step; (4) stage 1 of 10th load step; (5) stage 2 of 10th load step, and (6) 5th load step; NR-PCG = TNR.	20

List of Tables

Table 1. Approximate Hessian computed from previous iteration gradient information.....	13
Table 2. Rank-2 update approximation of Hessian inverse.	13

Acknowledgments

Yu Liang and Ramdev Kanapady are very pleased to acknowledge support in part by the U.S. Army High Performance Computing Research Center (AHPCRC) under the auspices of the U.S. Army Research Laboratory (ARL), Aberdeen Proving Ground, MD, contract no. DAAD19-01-2-0014. Special thanks are due for additional support to the Computational and Information Sciences Directorate's High Performance Computing Division at ARL. Other related support in the form of computer grants from the Minnesota Supercomputer Institute (MSI), Minneapolis, MN, is also gratefully acknowledged.

1. Introduction

The quasicontinuum (QC) method is commonly employed to solve a wide variety of multiscale problems (1–6). Typically, problems of interest are those involving a small number of defects such as dislocations, a crack, or grain boundary interactions (7). Unconstrained optimization constitutes the key computational kernel of this method (8–11). In this research, the objective is to study minimization solvers in order to optimize computational performance. Due to the implicit solution nature of the QC method and the predominant nonconvex nature of the potential energy surface with vast numbers of metastable configurations, one needs to use an effective and efficient iterative minimization solution technique. The efficiency of the techniques depends on the time needed to evaluate the energy expression and the number of iterations needed to converge to the minimum. In this context, a minimization iteration is defined as an iteration when the direction vector is updated. Iterations should not be confused with function evaluations nor the iterations involved with the linear equations solvers within the minimization framework. In this report, iterative solution techniques are explored for the local/nonlocal variant of the QC method (4) for the two-dimensional (2-D) situations. The fully nonlocal QC method with variable clusters (12) will be employed in a companion report for three-dimensional (3-D) situations.

Three popularly used unconstrained optimizing methods are steepest descent (SD), nonlinear conjugate gradient (NCG), and Newton-Raphson (NR) methods. Compared to the SD method (13), NCG methods (8, 14, 15) and the NR method (16) converge considerably faster. Compared to the NCG method, the NR method converges even faster as it employs the curvature of the objective function, in addition to the gradient information, to predict a search direction during the minimization iterations. However, due to the use of the second derivative of an objective function (i.e., Hessian matrix), the NR method suffers from two drawbacks. First, in each iteration, the search direction is obtained by solving the Newton equations, which have $\mathcal{O}(N^3)$ computational complexity when employing factorization-based direct solvers. Secondly, the Hessian is a dense matrix where the storage complexity is $\mathcal{O}(N^2)$. Alternatively, variants of the Newton method are available such as the quasi-Newton (QNR) and the truncated Newton methods (TNR). In TNR, a truncated strategy is used such that only an approximate solution to the Newton equations (16–25) is provided. The basic idea behind the QNR method follows from the NCG method by employing the gradients information of previous minimization iterations within the Newton framework, thereby avoiding a Hessian calculation. The TNR method differs from the QNR method in two aspects. First, an iterative method is employed for the solution of the line search direction. Second, the Hessian is not computed from previous gradient information.

Among these four minimization methods, the NCG and the NR methods are currently implemented for the local/nonlocal variant of the QC method found at qcmethod.com for 2-D situations. In this research, we examine the performance differences for two additional solver methods. We report the effectiveness of the TNR method with the conjugate gradient method for truncating the search direction and QNR method with low-rank Hessian update strategies used for computing the Hessian that are evaluated against the NR and the NCG methods. Results of illustrative examples mainly focus on the number of iterations and CPU time for the 2-D nanoindentation and shearing grain boundary problems.

The remainder of this report is organized as follows. In section 2, we review the basic equations and concepts related to the local/nonlocal variant of the QC formulation for the calculation of gradients and Hessians. In section 3, the four iterative solution techniques considered in this report are overviewed. Through these solvers, the objective is to determine the stable equilibrium configurations of a deforming crystalline material. In section 4, the result findings of the TNR method and QNR with low-rank Hessian update strategy are evaluated against the NR and the NCG implementation. Concluding remarks in section 5 follow.

2. Quasicontinuum Method

The two key components required for the minimization process are the gradient and Hessian computations. Over an energy landscape, the minima are identified as the lowest local point. When the state of the system is not co-located with a minimum, in a linear strategy, the minimum can be sought by following a combination of the local direction of the topology (i.e., the gradient and its local curvature, the Hessian). The gradient and Hessian are directly determined from the respective first and second gradients of an energy function. In this section, the local/nonlocal QC formulation is overviewed as it provides the energy function for our work. In section 2.5, the gradient and Hessian specific to this function is presented. As this report is to be self-contained, only those elements of the QC formulation vital to our discussion are presented. For further details, readers are referred to Shames and Dym (4).

2.1 Energy Framework

Figure 1a shows the computational framework of a typical QC simulation. In each load step, both the load-increment and resultant deformation are considered finite but small. Therefore, a Lagrangian description is used to formulate the kinematic behavior of the crystal. The heart of the QC method is the formulation of the total potential energy as an ensemble of a function whose independent variables are atom and finite-element (FE) nodal coordinates. The ground state (or state of static equilibrium) at zero temperature is found by minimizing the total potential energy or, equivalently, finding the zero out-of-balance force values of the following:

$$\min_{\mathbf{u} \in \mathbb{R}^{3N}} \Pi(\mathbf{u}), \quad (1)$$

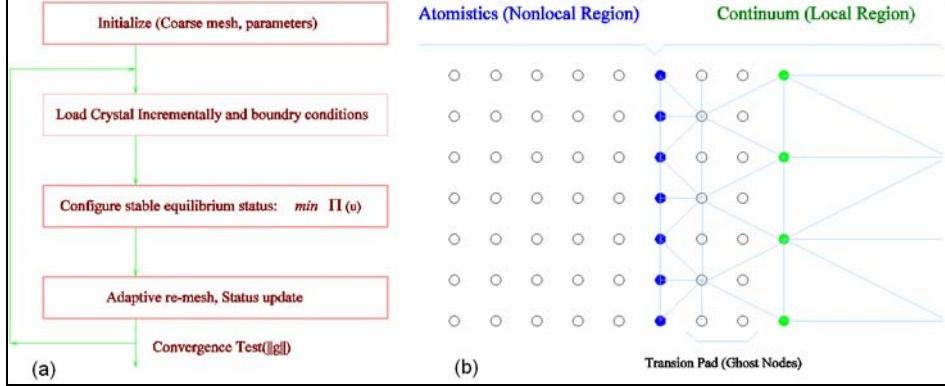


Figure 1. (a) Iterative process of quasicontinuum formulation and (b) local and nonlocal framework: black-filled circles are interface atoms shared by local and nonlocal regions; gray-filled circles are atoms selected as elemental nodes in local region; white-filled circles in the local region are formulated using kinematic constraints.

where the displacement of the representative atoms $\mathbf{u}(\mathbf{X}) = \mathbf{x} - \mathbf{X}$ is incurred by the incremental external force (\mathbf{f}_{ext}). Here, N is much smaller than the total number of degrees of freedom of the whole ensemble if the ensemble were comprised only of atoms, and \mathbf{X} and \mathbf{x} are the reference and deformed atomic coordinates, respectively. The total potential energy can be further represented as:

$$\Pi(\mathbf{u}) = E(\mathbf{u}) - \langle \mathbf{f}_{ext}, \mathbf{u} \rangle, \quad (2)$$

where E is the strain energy of the crystal. As illustrated in figure 1b, the crystal is divided into two regions—a local (continuum) region and a nonlocal (atomistic) region. Thus, the target strain energy function may be additively decomposed into local and nonlocal components as follows:

$$E(\mathbf{u}) = E_{\mathbf{X} \in \mathcal{L}}^{\mathcal{L}}(\mathbf{u}(\mathbf{X})) + E_{\mathbf{X} \in \{\mathcal{NL}, \mathcal{I}\}}^{\mathcal{NL}}(\mathbf{u}^{NL}(\mathbf{X}), \mathbf{u}^{\mathcal{I}}(\mathbf{X}), \mathbf{u}^{\mathcal{P}}(\mathbf{X})). \quad (3)$$

Here, \mathcal{L} , \mathcal{NL} , \mathcal{I} and \mathcal{P} are labels for the local region, nonlocal region, interface nodes, and transition area nodes (6), respectively. It is noted that $\mathcal{I} = \mathcal{L} \cap \mathcal{NL}$ and the transition area nodes provide a seamless coupling between local and nonlocal regions, and $\mathbf{u}^{\mathcal{L}}$, $\mathbf{u}^{\mathcal{NL}}$, $\mathbf{u}^{\mathcal{I}}$ and $\mathbf{u}^{\mathcal{P}}$ are the associated degrees of freedom of the regions \mathcal{L} , \mathcal{NL} , \mathcal{I} and \mathcal{P} , respectively. The first three, $\mathbf{u}^{\mathcal{L}}$, $\mathbf{u}^{\mathcal{NL}}$, and $\mathbf{u}^{\mathcal{I}}$ constitute the different components of \mathbf{u} , while $\mathbf{u}^{\mathcal{P}}$ is derived from $\mathbf{u}^{\mathcal{L}}$ by using kinematic constraints. $E^{\mathcal{L}}$ is computed at the element integration points and $E^{\mathcal{NL}}$ at the nodes. Figure 1b further illustrates that within the nonlocal (atomistic) region, a fully atomistic description is used to accurately depict the defect of the crystal. In addition, within the local (continuum) region, only a small number of atoms (representative -) are selected to describe the local material properties used in the FE method. In the end, it should be remarked that the partition of the ensemble is not fixed. An adaptive partition strategy is required to obtain an accurate description of the ensemble.

2.2 Atomistic Framework

In this work, the energy is formulated from the embedded atom method (EAM), which takes the atomic coordinates as direct input (17). In the nonlocal region, there is a one-to-one correspondence between nodes and atoms. Therefore, the energy of the nonlocal region is identical to a fully atomistic region. In the local region, where there is a smaller number of nodes than atoms, the energy over all atoms is determined by a kinematic approximation of atom deformation. By interpolating some of the atoms (i.e., those atoms that are not nodes), the strain energy is a locally constrained homogeneously deformed system. This kinematic approximation is known as the Cauchy-Born (CB) rule, which postulates that when a monatomic crystal is subjected to a small linear displacement of its boundary, then all atoms will follow this displacement (shown in figure 2).

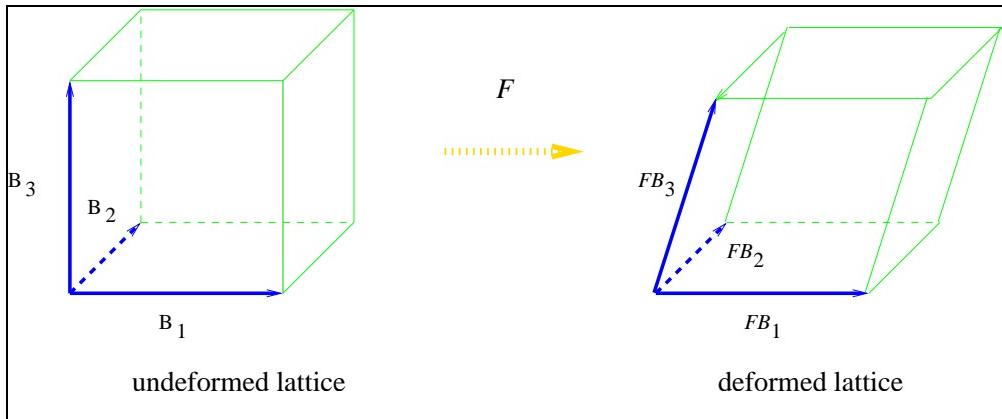


Figure 2. CB rule which transforms the reference lattice into homogeneously deformed (unified deformation gradient tensor \mathbf{F}) lattice.

Like other empirical atomistic models, $E^{\mathcal{NL}}$ can be approximated as the sum of the energies contributed from individual atomic sites (summation rule), i.e.,

$$E_{\mathbf{X} \in \{\mathcal{NL}, \mathcal{I}\}}^{\mathcal{NL}}(\mathbf{u}^{NL}(\mathbf{X}), \mathbf{u}^I(\mathbf{X}), \mathbf{u}^P(\mathbf{X})) = \sum_{\alpha \in \{\mathcal{NL}, \mathcal{I}\}} E_{\alpha}^{\mathcal{NL}}(\mathbf{u}^{NL}(\mathbf{X}), \mathbf{u}^I(\mathbf{X}), \mathbf{u}^P(\mathbf{X})). \quad (4)$$

Here, $E_{\alpha}^{\mathcal{NL}}$ is the site energy of α as computed using the EAM potential:

$$E_{\alpha}^{\mathcal{NL}}(\mathbf{u}) = \psi_{\beta}(\rho_{\beta}) + \psi_{\alpha}(\rho_{\alpha}) + \frac{1}{2} \sum_{\beta \neq \alpha} \vartheta_{\alpha}^{\beta}(\gamma_{\alpha, \beta}), \quad (5)$$

where

$$\rho_{\alpha} = \sum_{i \neq \alpha} f_{\alpha}^{\beta}(\mathbf{r}_{\alpha, \beta}) \quad (6)$$

and

$$\|\gamma_{\alpha, \beta}\| \leq \gamma_{cut}. \quad (7)$$

The embedding energy ψ_α is the energy associated with placing an atom in the electron environment described by local electron density ρ_α , and $\gamma_{\alpha,\beta} = \|\mathbf{x}_\alpha - \mathbf{x}_\beta\|$ is the interatomic distance. The pair-potential ϑ_α^i describes the electrostatic contributions of core ions, and γ_{cut} is the radius of the cutoff sphere. The cutoff is introduced to limit the number of modeled pair-wise interactions, which effectively truncates longer range interactions and accelerates the convergence of the sum. Assuming that there are m_β neighboring atoms inside the cutoff sphere of atom β , the deformed relative position of atom i near atom β is as follows:

$$\mathbf{r}_\beta^i = \mathbf{X}_\beta^i + \mathbf{u}_\beta^i - \mathbf{X}_\beta - \mathbf{u}_\beta . \quad (8)$$

If atom i is located in the transition or local region (shown in figure 2), \mathbf{u}_β^i is formulated using the following kinematic constraint:

$$\mathbf{u}_\beta^i = \sum_{\alpha=1}^N \mathbf{u}_\alpha \psi_\alpha(\mathbf{X}_\beta^i) . \quad (9)$$

Here, \mathbf{u}_α is the displacement of representative atom α , and ψ_α is the FE shape function. It follows that

$$E_{\mathbf{X} \in \{\mathcal{NL}, \mathcal{I}\}}^{\mathcal{NL}}(\mathbf{u}^{\mathcal{NL}}(\mathbf{X}), \mathbf{u}^{\mathcal{T}}(\mathbf{X}), \mathbf{u}^{\mathcal{P}}(\mathbf{X})) = \sum_{\beta=1}^{N_{NL}} n_\beta E_\beta^{NL}(\mathbf{r}_\beta^1, \dots, \mathbf{r}_\beta^{m_\beta}) , \quad (10)$$

where n_β is the number of atoms represented by atom β . Since a full atomistic strategy is employed to formulate the nonlocal region, we choose $n_\beta = 1$.

2.3 Continuum Framework

The strain energy related to local region $E_{\mathbf{X} \in \mathcal{L}}^{\mathcal{L}}(\mathbf{u}(\mathbf{X}))$ is computed by summing the potential energy of all elements within the local region as follows:

$$E_{\mathbf{X} \in \mathcal{L}}^{\mathcal{L}}(\mathbf{u}(\mathbf{X})) = \sum_{e=1}^M E_e^{\mathcal{L}} , \quad (11)$$

where

$$E_e^{\mathcal{L}} = \int_{\Omega_e} W_e(\mathbf{F}_e(\mathbf{X})) dV . \quad (12)$$

In these two formulae, M is the number of elements within the local region and W_e is the strain energy density depending on \mathbf{F}_e , a deformation gradient tensor of element e . Equation 12 is derived in accordance with the CB rule. Implied in equations 10 and 12 is that the present energy measures changes relative to the equilibrium lattice energy of the crystal such that the strain energy at zero deformation ($\mathbf{F}_e = \mathbf{I}$) in equilibrium is zero. The elemental deformation gradient tensor

$$\mathbf{F}_e = \frac{d\mathbf{x}}{d\mathbf{X}} = \mathbf{I} + \nabla \mathbf{u}(\mathbf{X}) \quad (\mathbf{X} \in \Omega_e) \quad (13)$$

describes the local and homogeneous stretching and rotating of element e . According to the principle of the FE method (26, 27), the displacement field of the element is formulated by using the following relation:

$$\mathbf{u}(\mathbf{X}) = \sum_{k=1}^{N_{ne}} \psi_k(\mathbf{X}) u_k, \quad (14)$$

where N_{ne} is the number of nodes per element, $\psi_k(\mathbf{X})$ is the shape function related to elemental node k , and u_k is the displacement of node k . Consequently,

$$\mathbf{F}_e(\mathbf{X}) = \mathbf{I} + \sum_{k=1}^{N_{ne}} \nabla \psi_k(\mathbf{X}) \otimes u_k, \quad (15)$$

where \otimes is the tensor product. Furthermore, using a specific numerical integration method, it follows from equation 11 that

$$E_{\mathbf{X} \in \mathcal{L}}^{\mathcal{L}}(\mathbf{u}(\mathbf{X})) == \sum_{e=1}^M v_e \mathcal{E}(\mathbf{F}_e), \quad (16)$$

which is evaluated at the quadrature points, and where v_e is the total number of atoms inside element e .

2.4 Coupling Atomistic and Continuum

The interface between local and nonlocal regions is nontrivial. For seamless joining between atomic and finite-element method (FEM) regions, the FEM region must have an internal atomic structure such that: (1) the assumption that strain in each element is homogeneous is reasonable, (2) elements have a scale, and (3) element corner nodes (r_i) are positioned on one of the atoms.

For each element, the work involved is as follows:

- Construct deformation gradient for this element.
- Pick a representative atom in the element.
- Calculate the energy per atom using the atomistic model.
- Assign energy to the element.

The link between atomistic and continuum material simulations requires a procedure for determining the nonlinear elastic continuum response of a material with a particular atomistic representation. The continuum response should match the results produced by discrete lattice

simulations at near atomistic length scales, while allowing more efficient simulation at larger scales through the application of the nonlinear continuum theory (28).

In general, particularly in the presence of lattice defects, the deformation cannot be assumed to be uniform at scales approaching lattice dimensions, even for infinitesimal deformations. The quasicontinuum procedure introduces degrees of freedom into the stored energy expressions which modify the computed constitutive properties to give better agreement with experimentally determined values. Following equations 10 and 16, it is derived that

$$\Pi_h(\mathbf{u}) = \sum_{e=1}^M v_e \mathcal{E}(\mathbf{F}_e) + \sum_{\beta=1}^{N_{NL}} E_{\beta}^{NL}(\mathbf{r}_{\beta}^1, \dots, \mathbf{r}_{\beta}^{m_{\beta}}) - \sum_{\alpha=1}^N n_{\alpha} \langle \mathbf{f}_{ext,\alpha}, \mathbf{u}_{\alpha} \rangle, \quad (17)$$

where n_{α} is the weight function related with representative atom α . In general, the interface between local and nonlocal atoms does not preserve force symmetry because atoms in local elements do not feel nearby nonlocal atoms, though nonlocal atoms do feel nearby local elements. As a remedy, the so-called ghost force correction is employed. To this end, the energy function is rewritten as follows:

$$\Pi_h(\mathbf{u}) = \sum_{e=1}^M v_e \mathcal{E}(\mathbf{F}_e) + \sum_{\beta=1}^{N_{NL}} E_{\beta}(\mathbf{r}_{\beta}^1, \dots, \mathbf{r}_{\beta}^{m_{\beta}}) - \sum_{\alpha=1}^N n_{\alpha} \langle \mathbf{f}_{ext,\alpha} + \mathbf{f}_{\alpha}^G, \mathbf{u}_{\alpha} \rangle. \quad (18)$$

2.5 Formulation of Gradient and Hessian

In order to obtain the equilibrium configuration of the solid, unconstrained minimization of the potential energy $\Pi_h(\mathbf{u})$ is required. Therefore, the corresponding gradient $\partial \Pi_h / \partial \mathbf{u}$ and sometimes Hessian $\partial^2 \Pi_h / \partial \mathbf{u}^2$ have to be computed.

2.5.1 Analytic Formulation of Gradient

Using the chain rule, it follows from equation 18 that

$$\frac{\partial \Pi_h}{\partial \mathbf{u}_{\alpha}} = \sum_{e=1}^M v_e P(\mathbf{F}_e) \frac{\partial \mathbf{F}_e}{\partial \mathbf{u}_{\alpha}} - \sum_{\beta=1}^{N_{NL}} \left[\sum_{j=1}^{m_{\beta}} \varphi_{\beta}^j \frac{\partial \mathbf{r}_{\beta}^j}{\partial \mathbf{u}_{\alpha}} \right] - n_{\alpha} (\tilde{\mathbf{f}}_{\alpha} + \mathbf{f}_{\alpha}^G), \quad (19)$$

where $P = \partial \boldsymbol{\varepsilon} / \partial \mathbf{F}$ is the first Piola-Kirchhoff stress tensor. Using

$$\mathbf{F}_e = \mathbf{I} + \sum_{\alpha=1}^R \mathbf{u}_{\alpha} \frac{\partial \psi_{\alpha}(\mathbf{X}_e)}{\partial \mathbf{X}} \mathbf{I}, \quad (20)$$

where \mathbf{X}_e is the element centroid, it follows that

$$\frac{\partial \mathbf{F}_e}{\partial \mathbf{u}_{\alpha}} = \frac{\partial \psi_{\alpha}(\mathbf{X}_e)}{\partial \mathbf{X}}. \quad (21)$$

Similarly,

$$\frac{\partial \mathbf{r}_{\beta}^j}{\partial \mathbf{u}_{\alpha}} = [\psi_{\alpha}(\mathbf{X}_{\beta}^j) - \delta_{\alpha\beta}] \mathbf{I}, \quad (22)$$

where $\delta_{\alpha\beta}$ is the Kronecker Delta. As a result, the out-of-balance nodal forces are computed by the following:

$$\mathbf{g}_\alpha = \frac{\partial \Pi_h}{\partial \mathbf{u}_\alpha} = \sum_{e=1}^M v_e P(\mathbf{F}_e) \nabla \psi_\alpha(\mathbf{X}_e) - \sum_{\beta=1}^{N_{NL}} \left[\sum_{j=1}^{m_\beta} \varphi_\beta^j \psi_\alpha(\mathbf{X}_\beta^j) \right] + \sum_{j=1}^{m_\beta} \varphi_\beta^j - n_\alpha (\tilde{\mathbf{f}}_\alpha + \mathbf{f}_\alpha^G), \quad (23)$$

where $P = \frac{\partial \Pi}{\partial F}$ is the first Piola-Kirchhoff stress tensor and \mathbf{X}_e is the element centroid.

2.5.2 Analytic Formulation of Hessian

Based on the gradient in (17), it can be further obtained that

$$\begin{aligned} \mathbf{G}_{\alpha\beta} &= \sum_{e=1}^M v_e C(\mathbf{F}_e) \nabla \psi_\alpha(\mathbf{X}_e) \nabla \psi_\beta(\mathbf{X}_e) + \sum_{\gamma=1}^{N_{h,NL}} \left[\sum_{k=1}^{m_\gamma} \sum_{l=1}^{m_\gamma} K_\gamma^{kl} \psi_\alpha(\mathbf{X}_\gamma^k) \psi_\beta(\mathbf{X}_\gamma^l) \right] \\ &\quad - \sum_{k=1}^{m_\alpha} \sum_{l=1}^{m_\alpha} K_\alpha^{kl} \psi_\beta(\mathbf{X}_\alpha^l) - \sum_{k=1}^{m_\beta} \sum_{l=1}^{m_\beta} K_\beta^{kl} \psi_\beta(\mathbf{X}_\beta^k) + \delta_{\alpha\beta} \sum_{k=1}^{m_\alpha} \sum_{l=1}^{m_\alpha} K_\alpha^{kl}, \end{aligned} \quad (24)$$

where $C = \frac{\partial^2 E}{\partial F^2}$ is the Lagrangian tangent stiffness tensor, $K_\beta^{kl} = \frac{\partial^2 E_\beta}{\partial \mathbf{r}_\beta^k \partial \mathbf{r}_\beta^l}$ is the atomic level stiffness matrix, and $P = \frac{\partial E}{\partial F}$ is the first Piola-Kirchhoff stress tensor. It is also useful to note that the general lack of symmetry in forces due to conjoined local/nonlocal domains, in the absence of correctively applied ghost forces, also translates to asymmetry in the Hessian. To demonstrate this, we refer to the following equations:

$$\mathbf{f}_L = \frac{\partial \mathcal{E}(\mathbf{F}_L)}{\partial \mathbf{F}_L} \frac{\partial \mathbf{F}_L}{\partial \mathbf{u}_L}, \quad (25)$$

and

$$\mathbf{f}_{NL} = \sum_{\beta=1}^n n_\beta \frac{\partial E_\beta}{\partial \mathbf{r}_\beta} \frac{\partial \mathbf{r}_\beta}{\partial \mathbf{u}_{NL}}. \quad (26)$$

It can be shown that the partial derivative $\partial \mathbf{f}_L / \partial \mathbf{u}_{NL} = \partial^2 \Pi_h / \partial \mathbf{u}_{NL} \partial \mathbf{u}_L$ is zero. However, the same is not true for the transpose

$$\frac{\partial \mathbf{f}_{NL}}{\partial \mathbf{u}_L} = \frac{\partial^2 \Pi_h}{\partial \mathbf{u}_L \partial \mathbf{u}_{NL}} = \sum_{\beta=1}^n n_\beta \frac{\partial \mathbf{r}_\beta}{\partial \mathbf{u}_L} \frac{\partial^2 E_\beta}{\partial \mathbf{r}_\beta \partial \mathbf{r}_\beta} \frac{\partial \mathbf{r}_\beta}{\partial \mathbf{u}_{NL}}, \quad (27)$$

where the only nonzero term in the summation is when $\beta = 1$. Since, in practice, $n_\beta = 1$, and based on the model in figure 2, $\mathbf{r}_1 = \mathbf{X}_L + \mathbf{u}_L - \mathbf{X}_{NL} - \mathbf{u}_{NL}$, we may simplify equation 18 as follows:

$$\partial^2 \Pi_h / \partial \mathbf{u}_L \partial \mathbf{u}_{NL} = -\partial^2 E_1(\mathbf{r}_1, \dots, \mathbf{r}_n) / \partial \mathbf{r}_1 \partial \mathbf{r}_1. \quad (28)$$

3. Solvers for Unconstrained Optimization

In this section, the TNR and QNR methods are presented. The NCG and the NR methods are also briefly reviewed.

In order to obtain the equilibrium configuration of the solid, unconstrained minimization of the total potential energy $\Pi(\mathbf{d})$ needs to be performed. From the discussions in the previous sections, we have gradient \mathbf{g} (equation 23), the Hessian \mathbf{H} (equation 24, referred as \mathbf{G} before), and approximate inverse (\mathbf{B}) of the Hessian. Then a general minimization iterative solver framework for unconstrained minimization can be described by algorithm 1. It is evident from this algorithm that if only the gradient is available, then QNR (29), steepest descent, and NCG (30, 31) are applicable. While the NR method has better convergence properties, the computation of an exact \mathbf{H} is time-consuming and may require large amounts of storage for large-scale problems. In algorithm 1, convergence is said to occur if $\|\mathbf{g}_k\| < \varepsilon$ is achieved.

•**Algorithm 1:** General iterative solver framework for minimization.

Initialization: \mathbf{u}_0 is given;

Relax Iteration:

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \alpha_k \mathbf{d}_k, \quad k = 0, \dots, \text{where } \mathbf{d}_k \text{ for:}$$

$$\text{Newton-Raphson: } \mathbf{d}_k = -(\mathbf{H}_k)^{-1} \mathbf{g}_k$$

$$\text{Quasi-Newton: } \mathbf{d}_k = -\mathbf{B}_k \mathbf{g}_k$$

$$\text{Steepest-descent: } \mathbf{d}_k = -\mathbf{g}_k$$

$$\text{Conjugate Gradient: } \mathbf{d}_k = -\mathbf{g}_k + \beta \mathbf{d}_k \text{ where } \beta \text{ for:}$$

$$\text{Fletcher-Reeves method: } \beta_k^{FR} = \left(\frac{\|\mathbf{g}_k\|_2}{\|\mathbf{g}_{k-1}\|_2} \right)^2$$

$$\text{Polak-Ribiere method: } \beta_k^{PR} = \frac{\langle \mathbf{g}_k, \mathbf{g}_k - \mathbf{g}_{k-1} \rangle}{\langle \mathbf{g}_{k-1}, \mathbf{g}_{k-1} \rangle}$$

$$\text{or Hestenes-Stiefel method: } \beta_k^{HS} = \frac{\langle \mathbf{g}_k, \mathbf{g}_k - \mathbf{g}_{k-1} \rangle}{\langle \mathbf{d}_{k-1}, \mathbf{g}_k - \mathbf{g}_{k-1} \rangle}$$

3.1 Nonlinear Conjugate Gradient Method

The nonlinear conjugate gradient method is of the following form:

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \alpha_k \mathbf{d}_k, \quad (29)$$

where $\alpha_k > 0$ is the step length and \mathbf{d}_k is search direction. Normally, the search direction at the first iteration is computed employing the steepest descent direction, namely, $\mathbf{d}_0 = -\mathbf{g}_0$. The other search directions can be defined recursively as follows:

$$\mathbf{d}_{k+1} = -\mathbf{g}_{k+1} + \beta_k \mathbf{d}_k. \quad (30)$$

Nonlinear conjugate gradient methods use search directions that combine the negative gradient direction with another direction, chosen so that the search will take place along a direction not previously explored by the algorithm. For the linear problems, the function minimizer is found exactly within just n iterations, thus performance is always predictable. For nonlinear problems, performance is problem dependent, but these methods have the advantage that they require only gradient evaluations. Memory requirements are minimal, making this a popular class of algorithms for large-scale optimization. Among common implementations, such as the Fletcher-Reeves, Polak-Ribiere, and Hestenes-Stiefel methods, the Polak-Ribiere nonlinear conjugate method appears to have the best convergence performance (described in algorithm 2).

•**Algorithm 2:** Polak-Ribiere conjugate gradient with given tolerance ε :

- (1) Initialize $\mathbf{u}_0; \mathbf{d}_0 = -\mathbf{g}_0;$
- (2) FOR $(k = 1, \dots, k_{\max})$ DO
- (3) Compute α such that $\min \Pi(\mathbf{u}_{k-1} + \alpha \mathbf{d}_{k-1}); /*$ line search */
- (4) $\mathbf{u}_k = \mathbf{u}_{k-1} + \alpha \mathbf{d}_{k-1};$
- (5) IF $(\|\mathbf{g}_k\| \leq \varepsilon)$ THEN exit;
- (6) $\beta_k = \frac{\langle \mathbf{g}_k, \mathbf{g}_k - \mathbf{g}_{k-1} \rangle}{\langle \mathbf{g}_{k-1}, \mathbf{g}_{k-1} \rangle}; /*$ Polak-Ribiere method */
- (7) $\mathbf{d}_k = -\mathbf{B}_k \mathbf{g}_k + \beta_k \mathbf{d}_{k-1};$
- (8) ENDFOR

3.1.1 Line Search Algorithm

Define $\Pi(\alpha) = \Pi(\mathbf{u} + \alpha \mathbf{d})$ where \mathbf{d} and \mathbf{u} are given. This potential energy can be redefined as a single variable function as $\phi(\alpha) = \Pi(\mathbf{u}_k + \alpha \mathbf{d}_k)$, which transforms $\min_x \Pi(x)$ into the minimizing problem $\min_{\alpha \in [\ell, h]} \phi(\alpha)$. Commonly used line-search methods are the NR method (where \mathbf{H}_k is needed) and the backtracking approach (9).

3.1.2 Backtracking

The step length control in the line search algorithm is mainly governed by the Wolfe conditions, which first requires that to avoid over-approximating the step length $\Pi(u_k + \alpha d_k) \leq \Pi(u_k) + c_1 \alpha \langle g_k, d_k \rangle$, where $c_1 = 10^{-4}$, which is based on the sufficient decrease condition. Secondly, to avoid under-approximating the step length, a condition is placed on the curvature and is given by $\langle g(u_k + \alpha_k d_k), d_k \rangle \geq c_2 \langle g_k, d_k \rangle$, where $c_2 = 0.9$. A backtracking line-search method is described in algorithm 1.

•**Algorithm 3:** Backtracking line search: given $\alpha_0 > 0$ and γ_1, γ_2 ($0 < \gamma_1 < \gamma_2 < 1$)

- (1) $k = 0;$
- (2) While $\Pi(\mathbf{u}_k + \alpha_k \mathbf{d}_k) > \Pi(\mathbf{u}_k) + c_1 \alpha_k \langle \mathbf{g}_k, \mathbf{d}_k \rangle$
- (3) Compute $\alpha_{k+1} \in [\gamma_1 \alpha_k, \gamma_2 \alpha_k]$ such that Wolfe conditions are satisfied;
- (4) $k = k + 1;$
- (5) ENDWhile

Statement (3) in algorithm 3 is implemented via the bisection method, gold section, or polynomial interpolation method (9). In our study, three point cubic interpolation methods are employed and are described briefly. Assume $\phi_c(\alpha) = a\alpha^3 + b\alpha^2 + c\alpha + d$ that satisfy $\phi_c(0) = \phi(0)$, $\phi'_c(0) = \phi'(0)$, $\phi_c(\alpha_k) = \phi(\alpha_{k-1})$, and $\phi_c(\alpha_k) = \phi(\alpha_k)$. Then, $\phi_c(\alpha) = a\alpha^3 + b\alpha^2 + \phi'(0)\alpha + \phi(0)$, where

$$\begin{bmatrix} a \\ b \end{bmatrix} = \theta \begin{bmatrix} -\alpha_k^2 \\ -\alpha_{k-1}^3 \alpha_k^3 \end{bmatrix} \begin{bmatrix} \phi(\alpha_k) - \phi(0) - \phi'(0)\alpha_k \\ \phi(\alpha_{k-1}) - \phi(0) - \phi'(0)\alpha_{k-1} \end{bmatrix} \quad (31)$$

and $\theta = 1/(\alpha_{k-1}^2 \alpha_k^2 (\alpha_k - \alpha_{k-1}))$. Using $\phi'_q(\alpha) = 0$ such that $\phi_c(\alpha)$ is minimized, it follows that $\alpha_{k+1} = \frac{b - \sqrt{b^2 - 3a\phi'(0)}}{3a}$.

3.2 NR Methods

For efficient and robust implementation of the NR methods, the following issues are critical:

- To make the method converge toward a local minimizer, the Hessian $\mathbf{G}^{(k)}$ has to be positive definite to guarantee the search direction $\mathbf{d}^{(k)}$ a descent direction, i.e., $\langle \mathbf{g}^{(k)}, \mathbf{d}^{(k)} \rangle < 0$. The NR method is not necessarily globally convergent, meaning that it may not converge from any starting point. The current guess $\mathbf{u}^{(0)}$ should be sufficiently close to \mathbf{u}^* so that the truncation error in equation A-1 of the appendix is negligible. In such case, $\mathbf{u}^{(k)}$ will converge to \mathbf{u}^* at a quadratic rate, i.e., $\|\mathbf{u}^{(k+1)} - \mathbf{u}^*\| \leq \varepsilon \|\mathbf{u}^{(k)} - \mathbf{u}^*\|^2$ where ε is a positive constant.
- In large-scale problems, the computation and storage requirements of the Hessian may become substantial. This can be handled by either using the diagonal terms in the Hessian, i.e., ignoring the cross terms, or avoiding recalculation at each iteration (can be done in instances of slow variation of the second derivative).
- The backtracking line-search scheme is recommended in the Newton-Raphson algorithm, i.e., in the full Newton step. $\alpha^{(k)}$ is tried first. If these steps fail to satisfy the criterion for the decrease of the function, backtrack in a systematic way along the Newton direction. The advantage of doing this is that fast convergence of the NR method will be obtained as the solution gets closer to the minimum.

3.2.1 Numerical Formulation of Hessian

Figure 3 illustrates several strategies in computing the Hessian (or the inverse of Hessian) (16–25), which are categorized into three branches of numerical methods. These are the analytic method, the difference method, and the low-rank update method. The formulation of the analytic (exact) Hessian is given by equation 24. In this section, we discuss only the low-rank update method.

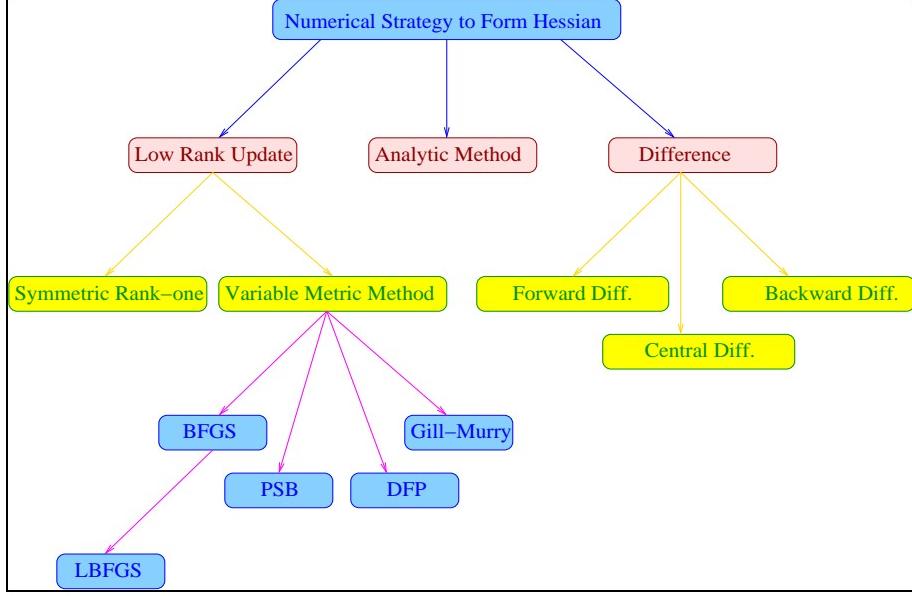


Figure 3. Numerical methods to formulate Hessian matrix.

3.2.2 Low-Rank Update Method

A low-rank update method (10, 32, 33) builds up an approximation to the Hessian by keeping track of the gradient differences along each step taken by the algorithm. Various conditions are imposed on the approximate Hessian. For example, its behavior along the step just taken is forced to mimic the behavior of the exact Hessian, and it is usually kept positive definite. Given $B^{(k)}$ as an approximation to the Hessian $G(u^{(k)})$, the low-rank update method iteratively updates this matrix by incorporating the curvature of the problem measured along the step according to the following secant condition (or Quasi-Newton condition): $B^{(k+1)}s^{(k)} = y^{(k)}$, where $y^{(k)} = g^{(k+1)} - g^{(k)}$, $s^{(k)} = u^{(k+1)} - u^{(k)}$. Table 1 lists four commonly used low-rank update methods. Broyden-Fletcher-Goldfarb (BFGS) is generally considered to be the most effective variable metric method.

In many applications, an inverse approximate Hessian $H^{(k)} \approx G^{(k)^{-1}}$ is more commonly used. Table 2 lists four commonly used rank-2 approximate inverse Hessians, which are formulated subject to the following secant condition: $H^{(k+1)}g^{(k)} = s^{(k)}$.

3.2.3 Convergence of the NR Method Based on Iterative Solver

Let $\{u^{(k)}\}$ be a sequence converging to u^* . The convergence about the associated gradient is linear, super-linear, and quadratic if there exists constants $\gamma \in (0, 1)$ and $\sigma > 0$ such that

$$\frac{\|g^{(k+1)}\|}{\|g^{(k)}\|} \leq \gamma, \quad \forall k, \quad (32)$$

$$\lim_{k \rightarrow \infty} \frac{\|g^{(k+1)}\|}{\|g^{(k)}\|} = 0, \quad (33)$$

Table 1. Approximate Hessian computed from previous iteration gradient information.

Broyden-Fletcher-Goldfarb (BFGS)	$\mathbf{B}^{(k+1)} \leftarrow \mathbf{B}^{(k)} + \frac{\mathbf{g}^{(k)} \times \mathbf{g}^{(k)}}{\langle \mathbf{g}^{(k)}, \mathbf{d}^{(k)} \rangle} + \frac{\mathbf{y}^{(k)} \times \mathbf{y}^{(k)}}{\langle \mathbf{y}^{(k)}, \mathbf{s}^{(k)} \rangle}.$
Powell-Symmetric (PSB)	$\mathbf{B}^{(k+1)} \leftarrow \mathbf{B}^{(k)} + \frac{(\mathbf{y}^{(k)} - \mathbf{B}^{(k)} \mathbf{s}^{(k)}) \times \mathbf{s}^{(k)} + \mathbf{s}^{(k)} \times (\mathbf{y}^{(k)} - \mathbf{B}^{(k)} \mathbf{s}^{(k)})}{\langle \mathbf{s}^{(k)}, \mathbf{s}^{(k)} \rangle} + \frac{(\mathbf{y}^{(k)} - \mathbf{B}^{(k)} \mathbf{s}^{(k)}) \times \mathbf{s}^{(k)}}{\langle \mathbf{s}^{(k)}, \mathbf{s}^{(k)} \rangle^2} (\mathbf{s}^{(k)} \times \mathbf{s}^{(k)}).$
Davidson-Fletcher-Powell (DFP)	$\mathbf{B}^{(k+1)} \leftarrow \mathbf{B}^{(k)} - \frac{(\mathbf{B}^{(k)} \mathbf{s}^{(k)}) \times (\mathbf{B}^{(k)} \mathbf{s}^{(k)})}{\langle \mathbf{s}^{(k)}, \mathbf{s}^{(k)} \rangle} \mathbf{B}^{(k)} + \frac{\mathbf{y}^{(k)} \times \mathbf{y}^{(k)}}{\langle \mathbf{y}^{(k)}, \mathbf{s}^{(k)} \rangle} + \langle \mathbf{s}^{(k)}, \mathbf{s}^{(k)} \rangle_{\mathbf{B}^{(k)}} (\mathbf{w}^{(k)} \times \mathbf{w}^{(k)}),$
where	$\mathbf{w}^{(k)} = \frac{\mathbf{y}^{(k)}}{\langle \mathbf{y}^{(k)}, \mathbf{s}^{(k)} \rangle} - \frac{\mathbf{B}^{(k)} \mathbf{s}^{(k)}}{\langle \mathbf{s}^{(k)}, \mathbf{s}^{(k)} \rangle}.$
SR1 (Symmetric Rank-1 Update)	$\mathbf{B}^{(k+1)} = \mathbf{B}^{(k)} + \frac{(\mathbf{y}^{(k)} - \mathbf{B}^{(k)} \mathbf{s}^{(k)}) \times (\mathbf{y}^{(k)} - \mathbf{B}^{(k)} \mathbf{s}^{(k)})}{\langle \mathbf{s}^{(k)}, \mathbf{y}^{(k)} - \mathbf{B}^{(k)} \mathbf{s}^{(k)} \rangle}.$

Table 2. Rank-2 update approximation of Hessian inverse.

BFGS	$\mathbf{H}^{(k+1)} \leftarrow \mathbf{H}^{(k)} + (\mathbf{I} - \frac{\mathbf{s}^{(k)} \mathbf{y}^{(k)\top}}{\langle \mathbf{s}^{(k)}, \mathbf{y}^{(k)} \rangle}) \mathbf{H}^{(k)} (\mathbf{I} - \frac{\mathbf{y}^{(k)} \mathbf{s}^{(k)\top}}{\langle \mathbf{s}^{(k)}, \mathbf{y}^{(k)} \rangle}) + \frac{\mathbf{s}^{(k)} \mathbf{s}^{(k)\top}}{\langle \mathbf{s}^{(k)}, \mathbf{y}^{(k)} \rangle}.$
PSB	$\mathbf{H}^{(k+1)} \leftarrow \mathbf{H}^{(k)} + \frac{(\mathbf{y}^{(k)} - \mathbf{H}^{(k)} \mathbf{s}^{(k)}) \times \mathbf{s}^{(k)} + \mathbf{s}^{(k)} \times (\mathbf{y}^{(k)} - \mathbf{H}^{(k)} \mathbf{s}^{(k)})}{\langle \mathbf{s}^{(k)}, \mathbf{s}^{(k)} \rangle} + \frac{(\mathbf{y}^{(k)} - \mathbf{H}^{(k)} \mathbf{s}^{(k)}) \times \mathbf{s}^{(k)}}{\langle \mathbf{s}^{(k)}, \mathbf{s}^{(k)} \rangle^2} (\mathbf{s}^{(k)} \times \mathbf{s}^{(k)}).$
DFP	$\mathbf{H}^{(k+1)} \leftarrow \mathbf{H}^{(k)} + (\mathbf{I} - \frac{\mathbf{s}^{(k)} \mathbf{y}^{(k)\top}}{\langle \mathbf{s}^{(k)}, \mathbf{y}^{(k)} \rangle}) \mathbf{H}^{(k)} (\mathbf{I} - \frac{\mathbf{y}^{(k)} \mathbf{s}^{(k)\top}}{\langle \mathbf{s}^{(k)}, \mathbf{y}^{(k)} \rangle}) + \frac{\mathbf{s}^{(k)} \mathbf{s}^{(k)\top}}{\langle \mathbf{s}^{(k)}, \mathbf{y}^{(k)} \rangle} - \mathbf{y}^{(k)\top} \mathbf{H}^{(k)} \mathbf{y}^{(k)} \mathbf{z} \mathbf{z}^\top.$
Gill and Murray	$\mathbf{H}^{(k+1)} \leftarrow \mathbf{H}^{(k)} - \frac{1}{\mathbf{y}^{(k)\top} \mathbf{s}^{(k)}} \left(\mathbf{H}^{(k)} \mathbf{y}^{(k)} \mathbf{s}^{(k)\top} + \mathbf{s}^{(k)} \mathbf{y}^{(k)\top} \mathbf{H}^{(k)} \right) + \frac{1}{\mathbf{y}^{(k)\top} \mathbf{s}^{(k)}} \left(1 + \frac{\mathbf{y}^{(k)\top} \mathbf{H}^{(k)} \mathbf{y}^{(k)}}{\mathbf{y}^{(k)\top} \mathbf{s}^{(k)}} \right) \mathbf{s}^{(k)} \mathbf{s}^{(k)\top}.$

and

$$\frac{\|\mathbf{g}^{(k+1)}\|}{\|\mathbf{g}^{(k)}\|^2} \leq \sigma, \quad \forall k. \quad (34)$$

Theorem 1 in the appendix shows that the truncated NR method converges at a super-linear rate.

3.2.4 Implementation of the NR Method

In our work, the NR method is utilized and described by algorithm 2.

•**Algorithm 4:** Newton-Raphson algorithm with given tolerance ϵ :

- (1) Initialize $u^{(0)}$
- (2) $d^{(0)} = -g^{(0)}$
- (3) FOR ($k = 0, \dots, k_{\max}$) DO
- (4) $\arg \min_{\alpha} \Pi(u^{(k)} + \alpha d^{(k)})$ /* line search */
- (5) $u_{k+1} = u_k + \alpha d_k$
- (6) update G_{k+1} and g_{k+1}
- (7) IF ($\|g_k\| \leq \epsilon$) THEN exit;
- (8) $G_{k+1}d_k = -g_{k+1}$ /* search direction */
- (9) ENDFOR

Two variants of NR strategies are: (1) a standard NR strategy where the Newton equations are solved using the direct method, i.e., LU factorization, and (2) a truncated NR method, where the Newton equations in statement (8) of algorithm 2 are solved approximately by preconditioned conjugate gradient algorithm (PCG).

•**Algorithm 5:** A preconditioned conjugate gradient solver for $Gd = g$, where the initial guess d_0 about the solution is given and C is the given preconditioner comprises algorithm 3.

- (1) $r_0 = g - Gd_0; z_0 = Cr_0; p_0 = z_0$
- (2) FOR $i = 0, 1, \dots$ till Convergence DO /* Krylov Iteration */
- (3) $\alpha_i = \frac{\langle r_i, z_i \rangle}{\langle Ap_i, p_i \rangle}$
- (4) $d_{i+1} = d_i + \alpha_i p_i$ /* update solution */
- (5) $r_{i+1} = r_i - \alpha_i A p_i$ /* update residual */
- (6) $z_{i+1} = Cr_{i+1}$ /* preconditioning */
- (7) $\beta_i = \frac{\langle r_{i+1}, z_{i+1} \rangle}{\langle r_i, z_i \rangle}$
- (8) $p_{i+1} = z_{i+1} + \beta_i p_i$
- (9) END FOR

Using the Hessian matrix, the optimal search direction can be obtained in the NR method. However, the $\mathcal{O}(R^2)$ memory requirements and $\mathcal{O}(R^3)$ associated with solving a linear system involving a Hessian restricts the NR methods only to (1) small-scale problems, (2) problems with special sparsity patterns, or (3) near a solution. As a remedy, Quasi-Newton, discrete Newton, and truncated Newton methods arise. The principle of these alternate methods is to obtain an approximation to the inverse of the Hessian matrix, i.e., $B^{(k)} \approx G^{-1}(u^{(k)})$, to obviate the difficulties that accompany its calculation and storage. One may also perform a nontrivial port of the NR method onto distributed computer systems using representative-atom-based, domain-decomposition methods. In light of these difficulties involving the Hessian, an out-of-core

truncated NR strategy is implemented and tested. This truncated Newton method is based on the idea that an exact solution of the Newton equation at every step is difficult and unnecessary and can be computationally wasteful in the framework of a basic descent method. The defining feature is that the search direction \mathbf{d} in the truncated Newton algorithm is computed using algorithm 6. While employing conjugate gradients (34) to approximate the search direction, the resulting convergence rate of algorithm 6 is strongly dependent on the condition number of $\mathbf{G}^{(k)}$. Among various methods, the preconditioning based on incomplete Cholesky factorization (34) has the best convergence performance. However, considering the large dimension of QC simulations and truncated nature of the search direction, only a diagonal-scaling preconditioner is often needed. This is the preconditioning approach employed in this work.

•**Algorithm 6:** Generating search direction \mathbf{d} in truncated Newton method:

```

(1)      Initialization:
(2)       $p_0 = 0; q_0 = 0; r_0 = -g$ 
(3)       $d_0 = C_0 r_0$ 
(4)      Iteration:
(5)      FOR ( $i = 0, \dots, i_{\max}$ ) DO
(6)          /* Section(1): negative curvature test */
(7)          IF ( $\langle d_i, d_i \rangle_G < \delta \langle d_i, d_i \rangle$ ) RETURN
(8)          /* Section(2): truncation test */
(9)           $\alpha_i = \frac{\langle C_i r_i, r_i \rangle}{\langle d_i, d_i \rangle_G}$ 
(10)          $p_{i+1} = p_i + \alpha_i d_i$ 
(11)          $r_{i+1} = r_i - \alpha_i G d_i$ 
(12)          $q_{i+1} = \frac{1}{2} \langle p_{i+1}, r_{i+1} + g \rangle$ 
(13)          $\|r_{i+1}\| < \epsilon$  RETURN
(14)         IF  $\beta_i = \frac{\langle C r_{i+1}, r_{i+1} \rangle}{\langle C r_i, r_i \rangle}$ 
(15)          $d_{i+1} = C_i r_{i+1} + \beta_i d_i$ 
(16)     ENDFOR

```

4. Numerical Experiments and Results

The deformation of crystalline materials is commonly studied based on lattice mechanisms and boundary mechanisms. In the lattice mechanisms, deformation occurs by processes taking place within the grains. In the boundary mechanisms (7), deformation occurs by processes associated with the presence of grain boundaries. Grain boundary sliding and diffusional creep are two major processes of boundary mechanisms. The NR, NCG method, QNR, and TNR methods are employed to simulate these two problems.

4.1 Shearing Grain Boundary

The grain boundary is comprised of two twins connected through an angled domain wall. The shearing boundary conditions are applied at the upper and lower boundaries. Lateral boundaries are connected through periodicity conditions. This problem is merely used as a test for convergence properties of the various solution algorithms. A grain boundary sliding process (7) employing the TNR method is demonstrated in figure 4. The solutions exactly match with the results of the NCG and NR methods found on qcmethod.com.

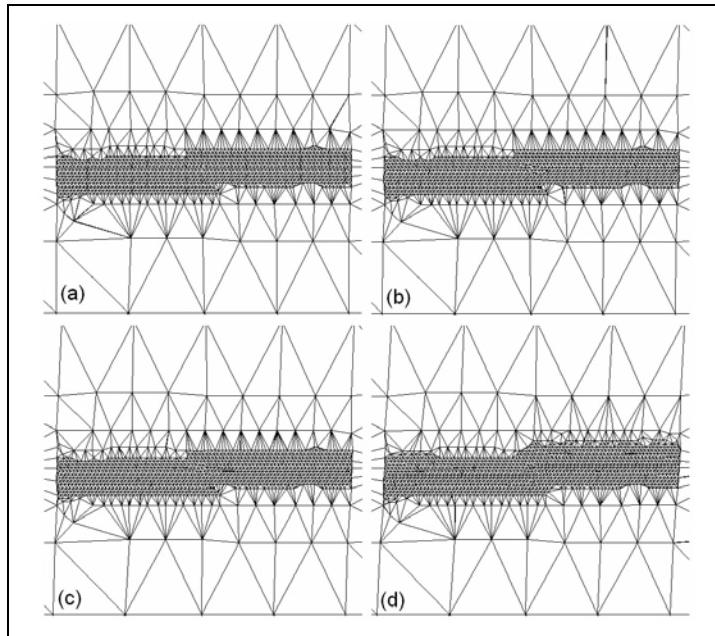


Figure 4. Sliding of grain boundary: (a) 1st load step, (b) 4th load step, (c) 7th load step, and (d) 11th load step.

Figures 5–7 compare the performance of the NR, NCG, and TNR methods. In this report, the NR method is based on LU factorization (direct solver, available in qcmethod.com) and PCG iterative solver (algorithm 3, implemented). A convergence tolerance for the PCG is said to be reached once the following condition is met:

$$\| \mathbf{r}_i \| / \| \mathbf{r}_0 \| \leq \epsilon, \quad (35)$$

where $\epsilon = 10^{-7}, 10^{-3}, 10^{-2}, 10^{-1}$ are employed in our numerical experiments for TNR method and \mathbf{r} is the residual in the PCG solution iterations. In our implementation, a diagonal scaling preconditioner is used to accelerate the convergence of PCG. Figure 5 shows that the number of required Newton iterations increases with the increase in the convergence criteria selected for the approximate search direction solver (PCG). In the limit, as the convergence criteria is decreased, the TNR method reduces to the NR method while, on the other hand, it will incur more inner

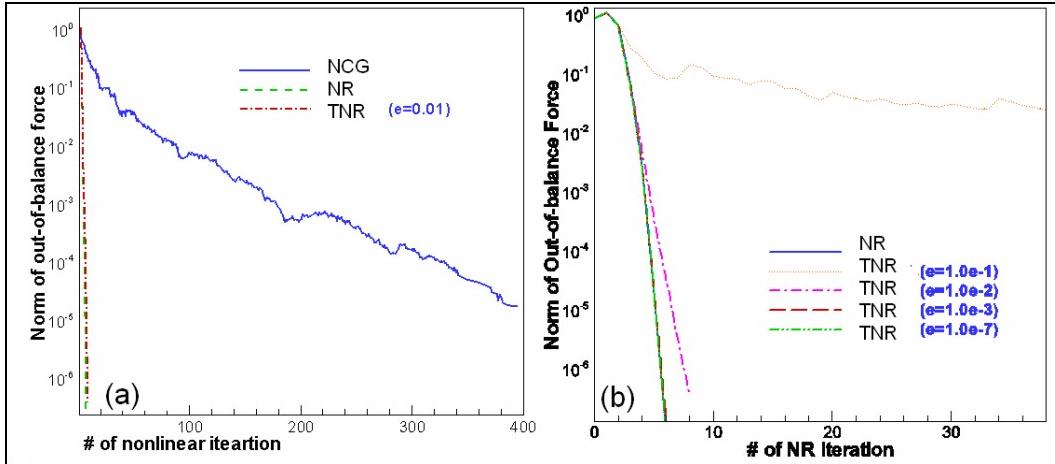


Figure 5. Number of minimization iterations for NCG, NR, and TNR for shear grain boundary problem for 10th load step.

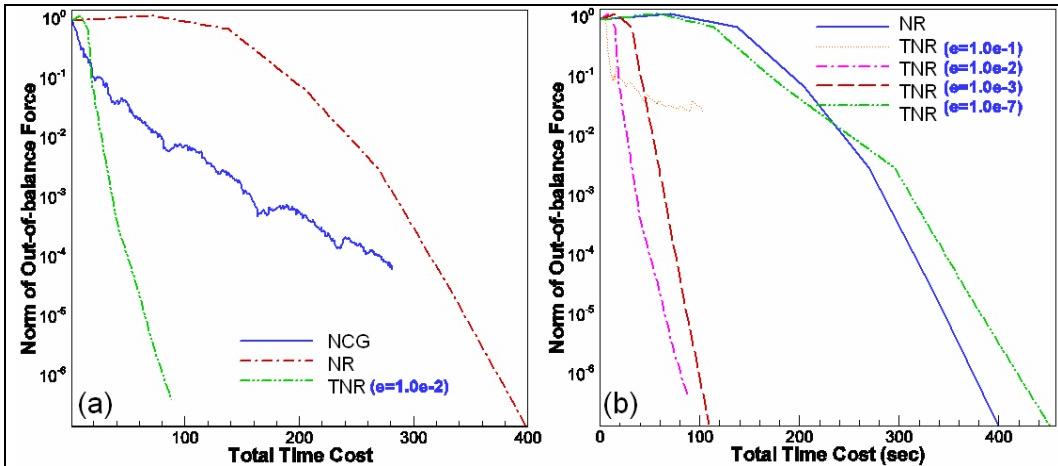


Figure 6. CPU time for NCG, NR, and TNR for shear grain boundary problem.

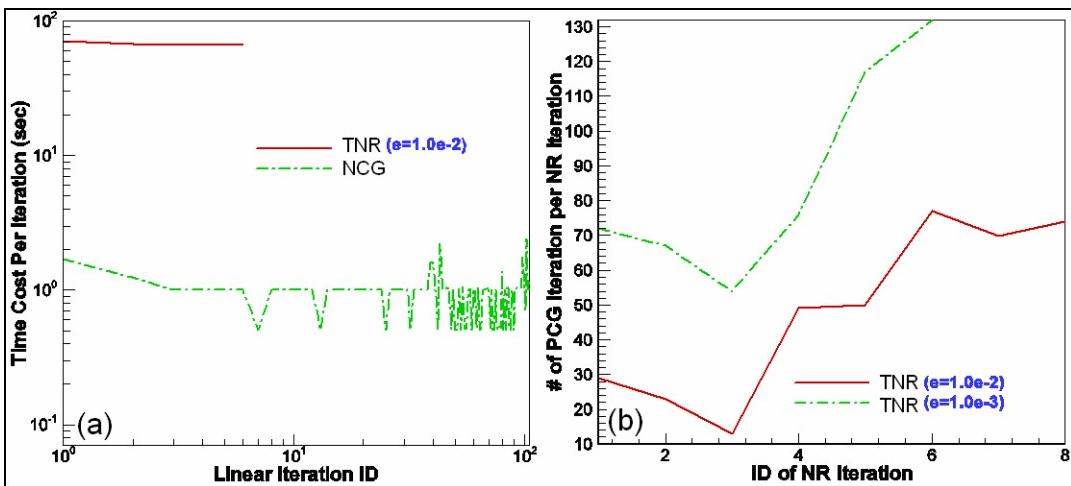


Figure 7. Time cost per minimizing iterations for TNR and NCG and number of iteration of linear conjugate gradient for the TNR for shear grain boundary problem.

PCG iterations. Figure 6a and b shows the CPU time for the four methods considered. It can be seen that for small PCG convergence tolerance (less than $\varepsilon = 10^{-7}$) and larger PCG convergence tolerances ($> \varepsilon = 10^{-1}$), the CPU time is greater than the NR method. For tolerance between $10^{-7} < \varepsilon < 10^{-1}$, the CPU time is less than the NR method. In all the cases, the CPU time is less than the NCG method. The NR and TNR methods reduce the number of nonlinear iterations by orders of magnitude. Figure 7 shows that the NR method takes about 66 s per iteration and NCG only takes 0.5–1.4 s per iterations.

4.2 Simulation of Nanoindentation

In nanoindentation, a nanometer scale indenter is pushed at constant speed from a given height to a given depth into the material (loading), and is subsequently retracted to its original position following the same path (unloading). For single crystals the measured load-displacement response, which shows the force required to push the tip a certain distance into the substrate, shows characteristic discontinuities. A sequence of deformation of the simulations is shown in figure 8. Figure 9 shows performance comparisons of the NCG, NR, and the QNR methods. From the figure, it is clear that the QNR performance is almost the same as the NCG method. Figure 10 shows the convergence performance of the NR, NCG, and TNR methods with respect

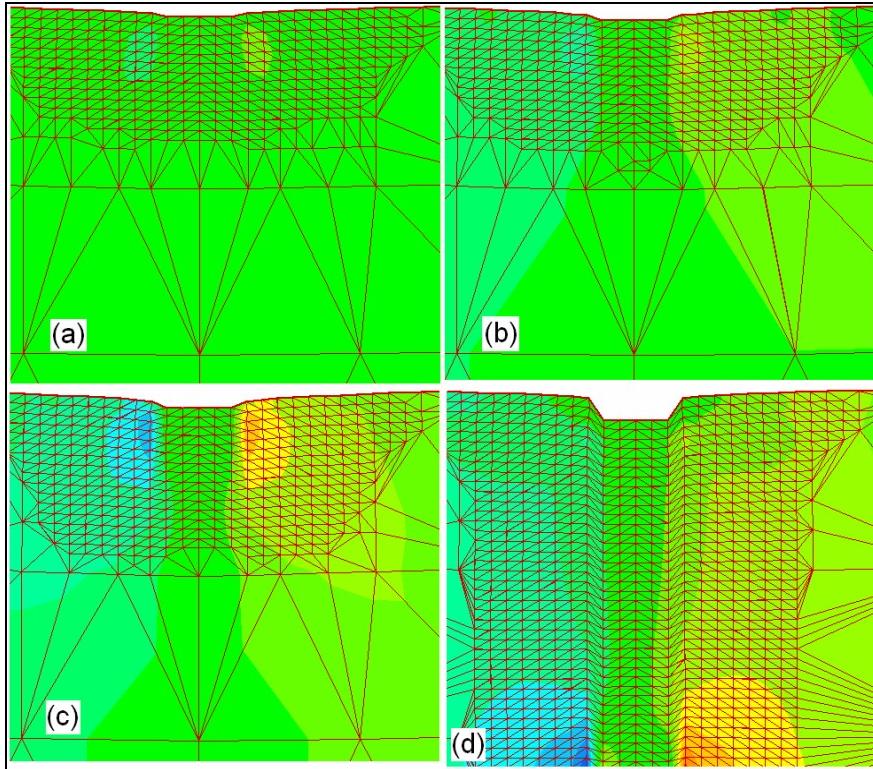


Figure 8. Nanoindentation problem: (a) 15th load step, (b) 20th load step, (c) 25th load step, and (d) 30th load step.

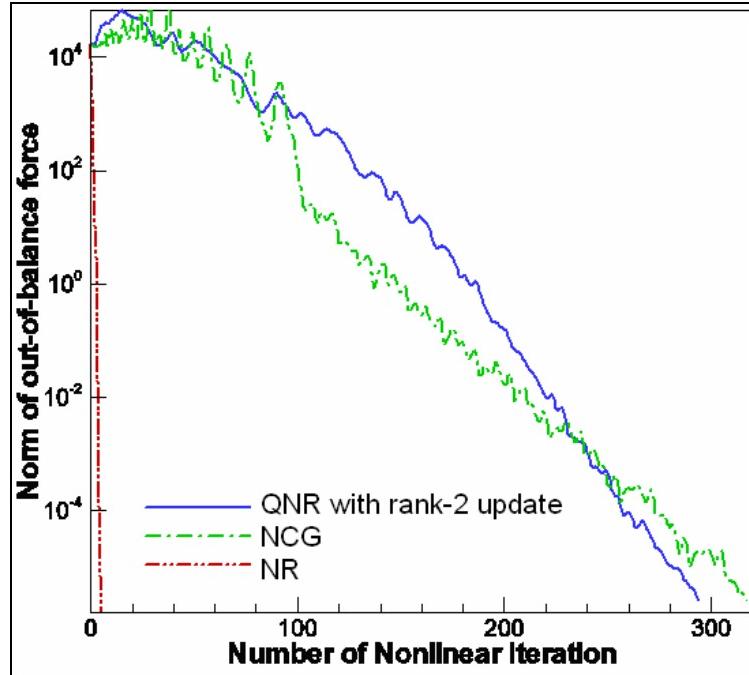


Figure 9. Performance comparison of the NCG, the NR, and QNR based rank-2 update Hessian for 10th load step of nanoindentation problem.

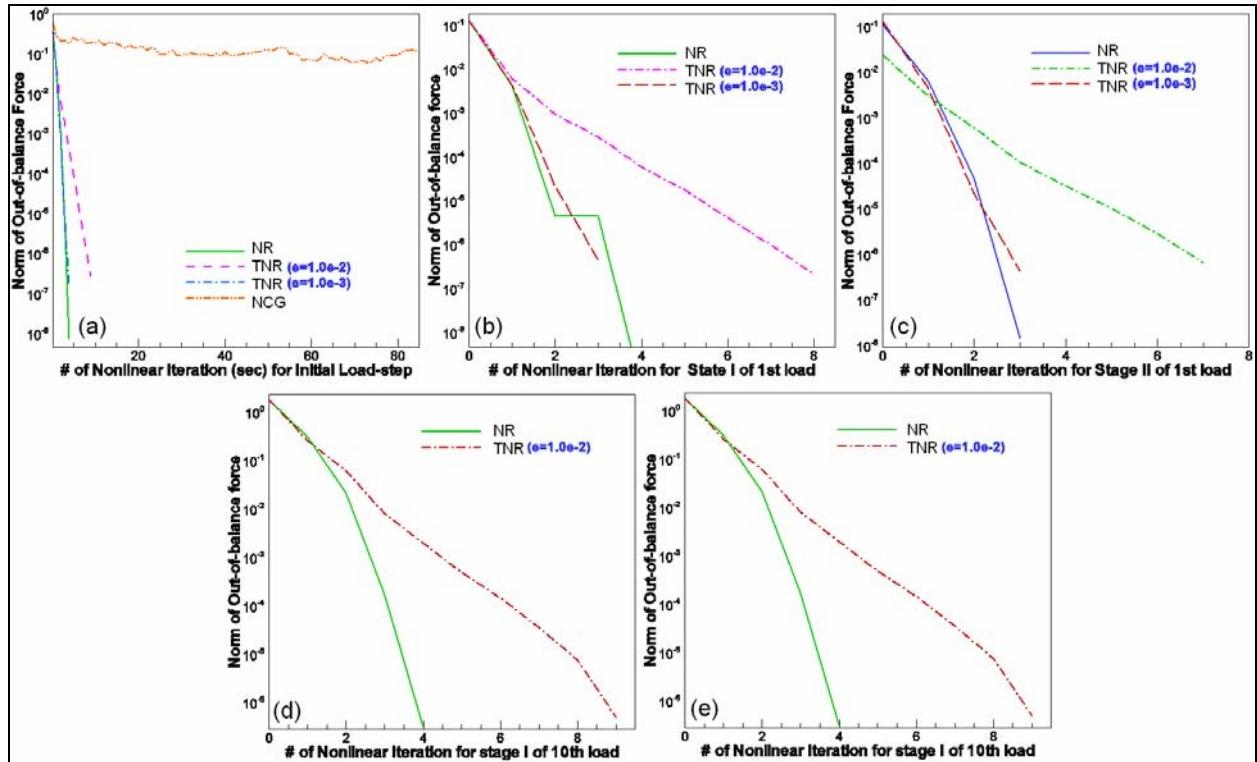


Figure 10. Convergence by nonlinear iteration for nanoindentation: (a) initial load step; (b) stage 1 of 1st load step; (c) 2nd load step; (d) 3rd load step; and (e) 4th load step; NR–PCG = TNR.

to various stages of the indentation process. It can be seen that NCG takes the largest number of iterations followed by TNR, with $\epsilon = 10^{-2}$ and with $\epsilon = 10^{-3}$ during the initialization stage, 1st load step and 10th load step. Further, figure 11 shows the CPU times of the NR and NCG methods for various load stages. It can be seen again that NCG takes most CPU time followed by the TNR method, with $\epsilon = 10^{-2}$ and with $\epsilon = 10^{-3}$ during the initialization stage, 1st load step and 10th load step. This is same performance as in the case of shear grain boundary problem considered previously.

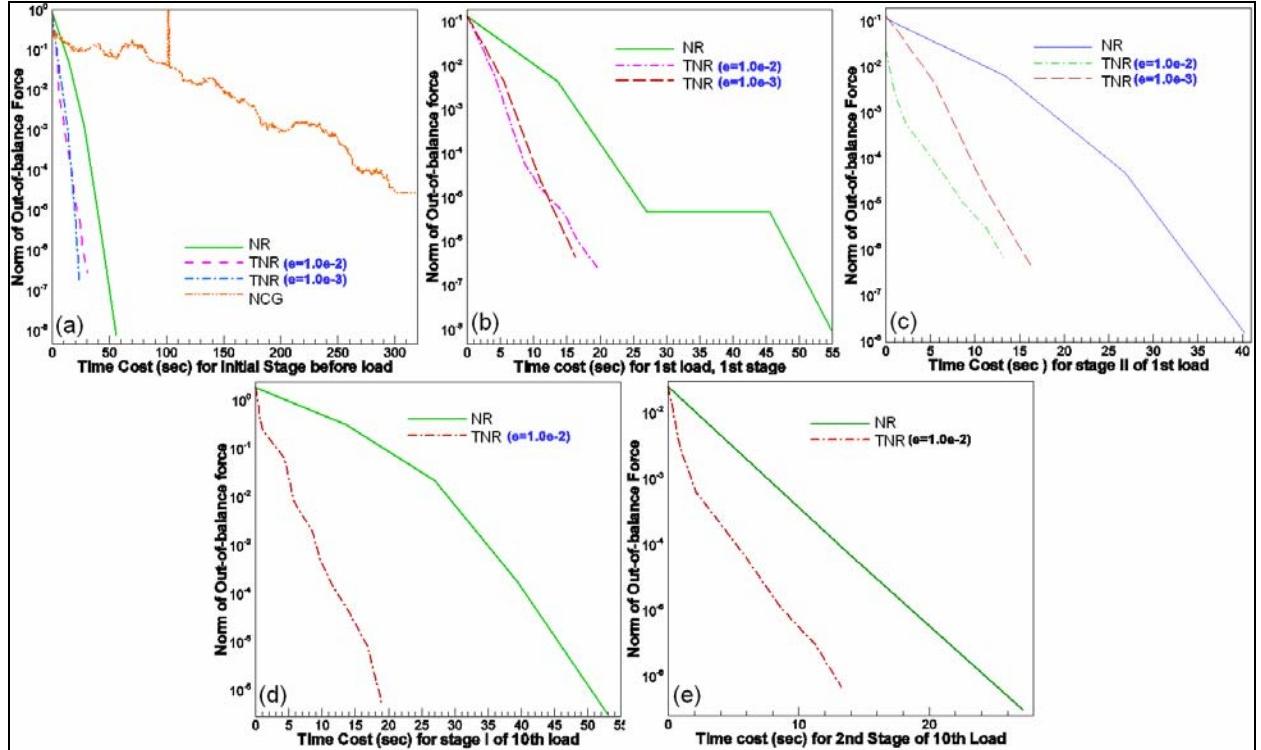


Figure 11. Convergence by timing cost (s) for nanoindentation: (1) initial load step; (2) stage 1 of 1st load step; (3) stage 2 of 1st load step; (4) stage 1 of 10th load step; (5) stage 2 of 10th load step, and (6) 5th load step; NR-PCG = TNR.

5. Conclusions

In the context of the two example problems studied in this report, we can reasonably conclude that the NR methods are more efficient than the NCG methods. However, the computational cost incurred for a full Newton method, in which the Hessian is recomputed and stored at every instance, can be prohibitive in general. A truncated Newton method may therefore be employed with a careful selection of the convergence tolerance such that the search directions are predicted with reasonable accuracy. Many complexities left unconsidered in this work still remain. First,

the nature of atomic forces and interactions are highly nonlinear with respect to atomic coordinates. Thus, in implicitly solving for quasistatic configurations of the material, the performance of matrix methods that are founded on linearization principles heavily depend on the condition of the material, namely its proximity to a minimum. Such approaches are therefore intrinsically unsuitable for capturing the “nonconvex” issues involved in atomistic problems. Indeed, one can identify states of deformation that result in instantaneous negative-definite configurations of atoms that violate the presumptions needed to employ linear solvers. Thus, the “region of viability” of linear solvers can be rather small in the overall energy landscape of possible solutions, and methods such as the proposed TNR method may suffer significant limitations in general problems.

6. References

1. Cleri, F.; et al. Atomistic Simulations of Materials Fracture and the Link between Atomic and Continuum Length Scales. *Journal of the American Ceramic Society* **1998**, *81* (3), 501–516.
2. Curtin, W. A.; Miller, R. E. Atomistic/Continuum Coupling in Computational Materials Science. *Modeling Simulation of Material Sci. and Eng.* **2003**, *11*, 33–88.
3. Knap, J.; Ortiz, M. An Analysis of the Quasicontinuum Method. *J. Mech. Phys. Solids* **2001**, *49*, 1899–1923.
4. Shames, I. H.; Dym, C. L. *Energy and Finite Element Methods in Structural Mechanics*; SI Unit Edition; Taylor & Francis: Philadelphia, PA, 1995.
5. Shenoy, V. B.; et al. An Adaptive Finite Element Approach to Atomic Scale Mechanics the Quasicontinuum Method. *Journal of the Mechanics and Physics of Solids* **1999**, *47*, 611–642.
6. Tadmor, E. B.; Ortiz, M.; Phillips, R. Quasicontinuum Analysis of Defects in Solids. *Philosophical Magazine A* **1996**, *73*, 1529–1563.
7. Beere, W. Stresses and Deformation at Grain Boundaries. *Phil. Trans. R. Soc. Lond. A* **288** *1978*, 177–196.
8. Dai, Y. H.; Yuan, Y. Full Text Available A Nonlinear Conjugate Gradient Method with a Strong Global Convergence Property. *SIAM Journal on Optimization* **1999**, *10* (1).
9. Dennis, J. E., Jr.; Mor'e, J. J. Quasi-Newton Methods, Motivation and Theory. *SIAM Rev.* **1977**, *19*, 46–89.
10. Golub, G. H.; Van Loan, C. F. *Matrix Computations*; 2nd ed.; The Johns Hopkins University Press: Baltimore, MD, 1993.
11. Nash, S. G.; Nocedal, J. A Numerical Study of the Limited Memory BFGS Method and the Truncated-Newton Method for Large-scale Optimization. *SIAM J. Optim.* **1991**, *1*, 358–372.
12. Gould, N. I. M.; Leyffer, S. An Introduction to Algorithms for Nonlinear Optimization; RAL-TR-2002-031; technique reports, 2002 (<http://www.numerical.rl.ac.uk/reports/reports.html>).
13. O'Leary, D. P. *Conjugate Gradients and Related KMP Algorithms: The Beginnings, in Linear and Nonlinear Conjugate Gradient-Related Methods*; Adams, L., Nazareth, J. L., Eds.; SIAM: Philadelphia, PA, 1996, pp 1–8.

14. Concus, P.; et al. *A Generalized Conjugate Gradient Method for the Numerical Solution of Elliptic Partial Differential Equations, Sparse Matrix Computation*; Bunch, J. R., Rose, D. J., Eds.; Academic Press: New York, NY, 1976.
15. Nocedal, J. Theory of Algorithms for Unconstrained Optimization. *Acta Numerica* **1992**, 199–242.
16. Dembo, R. S.; Eisenstat, S. C.; Steihaug, T. Inexact Newton Methods. *SIAM J. Num. Anal.* **1982**, 19, 400–408.
17. Daw, M.; Baskes, M. Semiempirical, Quantum Mechanical Calculation of Hydrogen Embrittlement in Metals. *Phys. Rev. Lett.* **1984**, 50, 1285–1288.
18. Dennis, J. E., Jr.; Schnabel, R. B. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*; Prentice-Hall, Inc.: Englewood Cliffs, NJ, 1983.
19. Miller, R. E.; Tadmor, E. B. The Quasicontinuum Method: Overview, Applications and Current Directions. *Journal of Computer-Aided Materials Design* **2002**, 9, 203–239.
20. Nash, S. G. Preconditioning of Truncated-Newton Methods. *SIAM J. Sci. Statist. Comput.* **1985**, 6, 599–616.
21. Papadrakakis, M.; Gantes, C. J. Preconditioned Conjugate-Newton and Secant-Newton Methods for Non-linear Problems. *International Journal for Numerical Methods in Engineering* **1989**, 28, 1299–1316.
22. Saad, Y. *Iterative Methods for Sparse Linear Systems*; PWS Publishing Co.: Boston, MA, 1996.
23. Schlick, T.; Fogelson, A. TNPACK A Truncated Newton Minimization Package for Large-Scale Problems. I: Algorithm and Usage, ACM. *Trans. Math. Software* **1992**, 18, 46–70.
24. Tadmor, E. B.; Phillips, R.; Ortiz, M. Mixed Atomistic and Continuum Models of Deformation in Solids. *Langmuir* **1996**, 12, 4529–4534.
25. Xie, D.; Schlick, T. Efficient Implementation of the Truncated Newton Method for Large Scale Chemistry Applications. *SIAM J. Opt.* **1999**, 10, 132–154.
26. Powell, M. J. D. Updating Conjugate Directions by the BFGS Formula; *Math. Programming* **1987**, 38, 693–726.
27. Schlick, T.; Fogelson, A. TNPACK A Truncated Newton Minimization Package for Large-Scale Problems. II: Implementation Examples, ACM. *Trans. Math. Software* **1992**, 18, 71–111.
28. Bonet, J.; Wood, R. D. *Nonlinear Continuum Mechanics for Finite Element Analysis*; Cambridge University Press: New York, NY, 2002.

29. Brown, P. N.; Saad, Y. Convergence Theory of Nonlinear Newton-Krylov Algorithms. *SIAM J. Opt.* **1994**, *4*, 297–330.
30. Broyden, C. G. Quasi-Newton Methods and Their Application to Function Minimization. *Math. Comp.* **1967**, *21*, 368–381.
31. Buckley, A.; LeNir, A. QN-like Variable Storage Conjugate Gradients. *Mathematical Programming* **1983**, *27*, 155–175.
32. Escudero, L. F. On Diagonally Preconditioning the Truncated Newton Method for Super Scale Linearly Constrained Nonlinear Programming. *European J. Oper. Res.* **1984**, *17*, 401–414.
33. Gill, P. E.; Leonard, M. W. Reduced-Hessian Quasi-Newton Method for Unconstrained Optimization. *SIAM J. Optim.* **2001**, *12* (1), 209–237.
34. Ross, C. T. F. *Advanced Applied Finite Element Methods*; Horwood Publishing Ltd.: Chichester, London, 1998.

Appendix. Supporting Derivations

A.1 Derivation of Newton-Raphson (NR) Method

The NR method is based on a quadratic approximation to the given function $\Pi(\mathbf{u})$ in each minimization iteration. The Taylor expansion yields the following:

$$\Pi(\mathbf{u}^{(k)} + \mathbf{p}^{(k)}) = \Pi(\mathbf{u}^{(k)}) + \mathbf{p}^{(k)\top} \mathbf{g}^{(k)} + \frac{1}{2} \mathbf{p}^{(k)\top} \mathbf{G}^{(k)} \mathbf{p}^{(k)} + \mathcal{O}(\|\mathbf{p}\|^3), \quad (\text{A-1})$$

where both gradient $\mathbf{g}^{(k)}$ and Hessian $\mathbf{G}^{(k)}$ are evaluated at $\mathbf{u}^{(k)}$. By ignoring the truncation error, it follows from equation A-1 that

$$\frac{\partial \Pi(\mathbf{u}^{(k)} + \mathbf{p}^{(k)})}{\partial \mathbf{p}^{(k)}} = \mathbf{g}^{(k)} + \frac{1}{2} \mathbf{G}^{(k)} \mathbf{p}^{(k)} + \frac{1}{2} \mathbf{G}^{(k)\top} \mathbf{p}^{(k)}. \quad (\text{A-2})$$

Mathematically, $\mathbf{G}^{(k)}$ is always symmetric if Π is twice continuously differentiable around $\mathbf{u}^{(k)}$, resulting in

$$\frac{\partial \Pi(\mathbf{u}^{(k)} + \mathbf{p}^{(k)})}{\partial \mathbf{p}^{(k)}} = \mathbf{g}^{(k)} + \mathbf{G}^{(k)} \mathbf{p}^{(k)}. \quad (\text{A-3})$$

According to the necessary condition of minimization, it is known that $\mathbf{G}^{(k)}$ is a symmetric positive definite (SPD) matrix. Assuming that $\mathbf{u}^{(k)} + \mathbf{p}^{(k)}$ is the minimizer, then

$$\frac{\partial \Pi(\mathbf{u}^{(k)} + \mathbf{p}^{(k)})}{\partial \mathbf{p}^{(k)}} = \mathbf{g}^{(k)} + \mathbf{G}^{(k)} \mathbf{p}^{(k)} = 0, \quad (\text{A-4})$$

which leads to the following Newton equation:

$$\mathbf{G}^{(k)} \mathbf{p}^{(k)} = -\mathbf{g}^{(k)}. \quad (\text{A-5})$$

As a result, NR is obtained by defining the search direction $\mathbf{d}^{(k)}$ as $\mathbf{d}^{(k)} = \mathbf{p}^{(k)} = -\mathbf{G}^{(k)^{-1}} \mathbf{g}^{(k)}$, and the step length $\alpha^{(k)}$ is set to be 1.0 by default due to the existence of truncation error in equation A-1. A line search is still required to find an appropriate $\alpha^{(k)}$.

A.2 Difference Hessian

The difference Hessian method is evaluated for a large scale three-dimensional situation in the companion report, which approximates the elements of the Hessian using a finite differencing technique, and is motivated by Taylor's theorem. When the second derivatives of objective function Π exists and are Lipschitz continuous, Taylor's theorem implies the following:

$$\nabla \Pi(\mathbf{u} + \mathbf{p}) = \nabla \Pi(\mathbf{u}) + \nabla^2 \Pi(\mathbf{u}) \cdot \mathbf{p} + \frac{1}{2} \mathbf{p} \cdot \nabla^3 \Pi(\mathbf{u}) \cdot \mathbf{p} + \mathcal{O}(\|\mathbf{p}\|^3), \quad (\text{A-6})$$

where “.” is scalar product. By substituting

$$\mathbf{p} = \tau \mathbf{e}_i, \quad (\text{A-7})$$

it follows that

$$\tau \nabla^2 \Pi(\mathbf{u}) \mathbf{e}_i = \nabla \Pi(\mathbf{u} + \tau \mathbf{e}_i) - \nabla \Pi(\mathbf{u}) - \frac{\tau^2}{2} \mathbf{e}_i \cdot \nabla^3 \Pi(\mathbf{u}) \cdot \mathbf{e}_i + \mathcal{O}(\tau^3). \quad (\text{A-8})$$

Similarly, let $\mathbf{p} = -\tau \mathbf{e}_i$ where $\tau > 0$ be a small increment such that we obtain the following:

$$\tau \nabla^2 \Pi(\mathbf{u}) \mathbf{e}_i = -\nabla \Pi(\mathbf{u} - \tau \mathbf{e}_i) + \nabla \Pi(\mathbf{u}) + \frac{\tau^2}{2} \mathbf{e}_i \cdot \nabla^3 \Pi(\mathbf{u}) \cdot \mathbf{e}_i + \mathcal{O}(\tau^3). \quad (\text{A-9})$$

According to equations A-8 and A-9, an approximate Hessian can be formulated using the following central difference method:

$$\mathbf{G}_{*,i} = \nabla^2 \Pi(\mathbf{u}) \mathbf{e}_i = \frac{\mathbf{g}(\mathbf{u}^{(k)} + \tau \mathbf{e}_i) - \mathbf{g}(\mathbf{u}^{(k)} - \tau \mathbf{e}_i)}{2\tau} + \mathcal{O}(\tau^2). \quad (\text{A-10})$$

Let \mathbf{B} be the approximation of \mathbf{G} by ignoring the truncation error, i.e.,

$$\mathbf{B}_{*,i} = \frac{\mathbf{g}(\mathbf{u}^{(k)} + \tau \mathbf{e}_i) - \mathbf{g}(\mathbf{u}^{(k)} - \tau \mathbf{e}_i)}{2\tau}. \quad (\text{A-11})$$

Then the difference Hessian method is easily formulated and suitable for parallel computing (each element of the Hessian is formulated independently). In addition, it only requires the numerical evaluation of the gradient \mathbf{g} .

Theorem 1: Convergence Rate of TNR: Assume that

1. $\mathbf{g}^{(k)} = \nabla \Pi(\mathbf{u}^{(k)})$ is continuously differentiable in a neighborhood of \mathbf{u}^* , a local minimizer of Π .
2. $\nabla^2 \Pi(\mathbf{u}^*)$ is nonsingular and that $\nabla^2 \Pi$ is Lipschitz continuous at \mathbf{u}^* . Then the truncated NR method (see references 16–25 in section 6 of this report), based on an iterative solver, has a super-linear convergence rate.
3. The truncated NR method is employed to find \mathbf{u}^* , i.e., the search direction $\mathbf{d}^{(k)}$ is obtained by solving the following Newton equation:

$$\mathbf{G}^{(k)} \mathbf{d}^{(k)} = -\mathbf{g}^{(k)} \quad (\text{A-12})$$

with the following convergence criterion:

$$\|\mathbf{r}_i^{(k)}\| \leq \varepsilon \|\mathbf{g}^{(k)}\|, \quad (\text{A-13})$$

where $\varepsilon_k \leq \varepsilon_{\max} < 1$ is the accuracy tolerance, then

$$\lim_{k \rightarrow \infty} \sup \frac{\|g^{(k+1)}\|}{\|g^{(k)}\|} \leq \varepsilon < 1. \quad (\text{A-14})$$

Proof: Let $\mathbf{G}^{(k)}$ be SPD and assume there exists a constant $\sigma > 1$ for all $\mathbf{u}^{(k)}$ that is sufficiently close to \mathbf{u}^* , $\|\mathbf{G}^{(k)}\|^{-1} \leq \sigma$. Thus, it follows from equations A-12 and A-13 that

$$d_i^{(k)} \leq \sigma(\|g^{(k)}\| + \|r_i^{(k)}\|) \leq (1 + \varepsilon)\sigma\|g^{(k)}\|. \quad (\text{A-15})$$

According to the Taylor's expansion about the gradient $g^{(k+1)} = g^{(k)} + d^{(k)}$, it follows that

$$g^{(k+1)} = g^{(k)} + \mathbf{G}^{(k)}d_i^{(k)} + \mathcal{O}(\|d_i^{(k)}\|^2) = r_i^{(k)} + \mathcal{O}((1 + \varepsilon)^2\sigma^2\|g^{(k)}\|^2). \quad (\text{A-16})$$

Thus, using equation A-13, it follows that

$$\|g^{(k+1)}\| \leq \varepsilon\|g^{(k)}\| + \mathcal{O}(\|g^{(k)}\|^2). \quad (\text{A-17})$$

As a result, if $\mathbf{u}^{(k)}$ is sufficiently close to \mathbf{u}^* , then equation A-14 holds.

NO. OF
COPIES ORGANIZATION

1 DEFENSE TECHNICAL
(PDF INFORMATION CTR
ONLY) DTIC OCA
8725 JOHN J KINGMAN RD
STE 0944
FORT BELVOIR VA 22060-6218

1 US ARMY RSRCH DEV &
ENGRG CMD
SYSTEMS OF SYSTEMS
INTEGRATION
AMSRD SS T
6000 6TH ST STE 100
FORT BELVOIR VA 22060-5608

1 INST FOR ADVNCED TCHNLGY
THE UNIV OF TEXAS
AT AUSTIN
3925 W BRAKER LN
AUSTIN TX 78759-5316

1 DIRECTOR
US ARMY RESEARCH LAB
IMNE ALC IMS
2800 POWDER MILL RD
ADELPHI MD 20783-1197

3 DIRECTOR
US ARMY RESEARCH LAB
AMSRD ARL CI OK TL
2800 POWDER MILL RD
ADELPHI MD 20783-1197

ABERDEEN PROVING GROUND

1 DIR USARL
AMSRD ARL CI OK TP (BLDG 4600)

NO. OF
COPIES ORGANIZATION

- 1 DIR USARL
AMSRD ARL SE E
N FELL
2800 POWDER MILL RD
ADELPHI MD 20783-1197
- 2 DIR USARL
AMSRD ARL SE EI
N DHAR
J LITTLE
2800 POWDER MILL RD
ADELPHI MD 20783-1197
- 4 DIR USARL
AMSRD ARL SE RL
P AMIRTHARAJ
L CURRENNO
J PULSKAMP
A WICKENDEN
2800 POWDER MILL RD
ADELPHI MD 20783-1197

ABERDEEN PROVING GROUND

- 22 DIR USARL
AMSRD ARL CI H
C NIETUBICZ
AMSRD ARL CI HC
J BLAUDEAU
J CLARKE
C CORNWELL
B HENZ
R NAMBURU
D SHIRES
A YAU
AMSRD ARL WM
M GREENFIELD
J MCCUALEY
J SMITH
T WRIGHT
AMSRD ARL WM BD
B FORCH
M HURLEY
AMSRD ARL WM MA
J ANDZELM
M VAN LANDINGHAM
AMSRD ARL WM TA
S SCHOENFELD
AMSRD ARL WM TD
T BJORKE
J CLAYTON
K IYER
B LOVE
S SEGLETES

INTENTIONALLY LEFT BLANK.